OGSA-DAI Status and Benchmarks

Mario Antonioletti, Malcolm Atkinson, Rob Baxter, Andrew Borley, Neil P. Chue Hong, Patrick Dantressangle, Alastair C. Hume, Mike Jackson, Kostas Karasavvas, Amy Krause, Simon Laws, Mark Parsons, Norman W. Paton, Jennifer M. Schopf, Tom Sugden, Kostas Tourlas, Paul Watson, and David Vyvyan

Abstract

This paper presents a status report on some of the highlights that have taken place within the OGSA-DAI project since the last AHM. In addition, a description of Release 6.0 functionality, due in April, and details of the forthcoming Release 7.0, due in September, is given as well as discussing some future directions for this project. In particular initial results of work done to systematically benchmark recent OGSA-DAI releases is presented. The OGSA-DAI software distribution and more information about the project is available from the project website at http://www.ogsadai.org.uk.

1 Present Status and Infrastructure

OGSA-DAI [1] is a widely used piece of middleware infrastructure that enables client applications to submit request documents in order to perform a set of activities on a remote data resource. The first release of OGSA-DAI took place in January 2003 and since then there have been 5 major and 3 minor releases with over 3900 downloads. OGSA-DAI has over 1100 registered users, including projects which require continuously available data access and integration services.

In the time since the 2004 AHM there have been several major changes within OGSA-DAI, the largest of these being a move to support three different underlying messaging infrastructures. Release 5.0 was the last release focussing solely on OGSI, with new features including content-based accessing of files. This came out in December 2004 and, by March 2005, had been downloaded over 600 times. Three versions of OGSA-DAI are now available based on: WS-I, as provided by the WS-I+ based OMII platform, WSRF, as implemented in the GT4.0, and OGSI, under GT3. By March technical previews of the WSRF and WS-I distributions were available from the Globus and OGSA-DAI web sites.

Significant refactoring of the code has been required to make maintenance and support of these multiple platforms tractable. A three-tiered implementation, shown in Figure 1, has now been adopted. The DAI core component serves the three different messaging frameworks. The WSRF and WS-I service-side components are now available. Development of the OGSI component is now deprecated but will still receive support up to release 7.0. A new client-side API, designed to abstract the differences between underlying infrastructures, will be available from release 6.0.

The WS-I version will be part of the OMII distribution from June 2005. OGSA-DAI is migrating from a Technical Preview to a Core Component of the Globus Toolkit.

Although it has been possible to support three different infrastructures, it has taken effort has from extending functionality and improving performance. It is hoped that in the near future, the community will be able to converge on a single infrastructure.

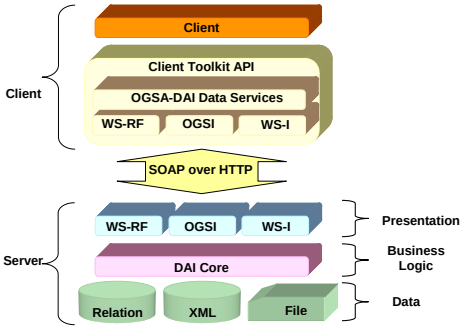

Figure 1: Schematic representation of the new OGSA-DAI tiered implementation**.**

2 Upcoming Features

With the refactoring to accommodate multiple infrastructures now completed, the focus has moved to adding functionality and improving usability and performance. Additions to Release 6.0 and 7.0 include:

- Dynamic service configuration: allowing services to be reconfigured without re-starting the container.
- Client toolkit API: providing a toolkit that hides the differences between infrastructures from applications.
- Benchmarking

- Closer integration with OMII and Globus Toolkit middleware.

A longer-term effort, led by Malcolm Atkinson, is reviewing and generalising the OGSA-DAI architecture starting from release 7.0 and beyond, [2]. The new architecture is currently being prototyped. Some of the concepts currently being undertaken include:

- Code mobility: dynamic insertion of third party activities.
- Transaction support: support atomic and distributed transactions.
- Engine redesign: support for concurrency and sessions.
- Control flow in perform documents.

3 Projects and Collaborations

New projects, in addition to those reported at the previous AHM meeting, [3], have adopted OGSA-DAI for their data access requirements, including: Cancer Biomedical Information Grid (cabig.nci.nih.gov) [4], Lead (www.lead.ou.edu), Data Mining Grid (www.datamininggrid.org), SIMDAT (www.simdat.org) and GOLD (www.goldproject.ac.uk). These projects are using the activity framework to tailor code to their own requirements and deploy it through OGSA-DAI. An analysis of the current use of OGSA-DAI with illustrations from a range of projects will be presented in the final paper.

The third OGSA-DAI users' group meeting, scheduled to take place at NeSC in early June will allow users to comment on OGSA-DAI and our planned developments. It will collect requirements assess priorities.

A number of major projects are currently consulting the team as part of their evaluation or as part of their design concerning their use of OGSA-DAI. A summary of these projects, as well as the process for future collaborative work on the OGSA-DAI software will be presented.

The OGSA-DAI team are in an e-Science sisters project, DIALOGUE, which is developing a strategy for combining multiple data integration systems. The team continues to contribute to standardisation and three GGF DAIS recommendation documents are planned by mid 2005.

4 Benchmarking OGSA-DAI

Although some effort has previously been expended by the OGSA-DAI team to do performance analysis and optimisation of the OGSA-DAI code [5], these have generally been one-off tasks. A more systematic approach is now being taken to produce a benchmark suite that will be run automatically. The recorded results will be made public to encourage selection of priorities and focused improvements.

Initial studies have shown that the performance of OGSA-DAI can vary markedly depending on which of the various activities and delivery mechanisms are used. For example, Figure 2 compares the performance of an OGSA-DAI query using three different delivery mechanisms.

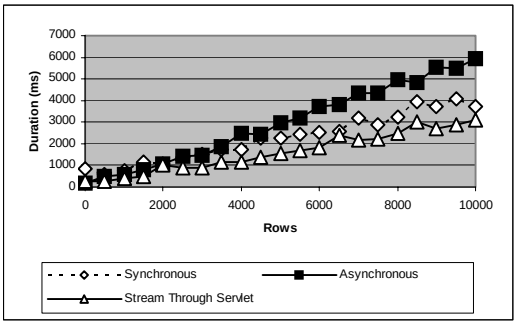

Figure 2: OGSA-DAI performance using synchronous, asynchronous, and servlet delivery.

The AHM paper will present initial performance results using a variety of activity and delivery methods. As well as being of importance to OGSA-DAI users, the results will also be of interest to those working on more general Web Service data transfer performance issues.

5 Conclusions

This abstract only presents a summary of the salient points to be covered in the full paper. Two years after its first release OGSA-DAI continues to mature and expand in its functionality. Both the user and contributor base are growing. They include major projects in the USA, Japan and China.

6 References

[1] M. Antonioletti, et al., *The design and implementation of Grid database services in OGSA-DAI*. Concurrency and Computation: Practice and Experience 17(2): 357-376.

[2] M. Atkinson, et al., *A New Architecture for OGSA-DAI*. Submitted to the AHM 2005.

[3] M. Antonioletti, et al., *OGSA-DAI Status Report and Future Direction*. Proceedings of the UK e-Science All Hands Meeting 2004, September 2004.

[4] W. Sanchez, et al., *caGRID White Paper*. July 23, 2004. See http://cabig.nci.nih.gov/guidelines_documentation/caGRIDWhitepaper.pdf.

[5] M. Jackson, et al., *Performance Analysis of the OGSA-DAI Software*. Proceedings of the UK e-Science All Hands Meeting 2004, September 2004.